

Javascript Web-Applikation als Docker Container in der Cloud verfügbar machen

Ausgangslage

Sie haben eine fertige Node JS-Webapplikation lokal auf ihrem Gerät und wollen diese nun in der Cloud verfügbar machen.

Zielgruppe

Lernende des dritten Lehrjahres Abteilung Informatik

Voraussetzungen

- Docker installiert
- Azure konfiguriert
- Kenntnisse mit Azure und Docker



Was ist nicht Bestandteil dieser Anleitung

- Auf Konfiguration von Docker und Azure Cloud wird nicht näher eingegangen
- Intergration der Datenbank ist nicht Bestandteil dieser Anleitung
- Port forwarding

Bitte beachten

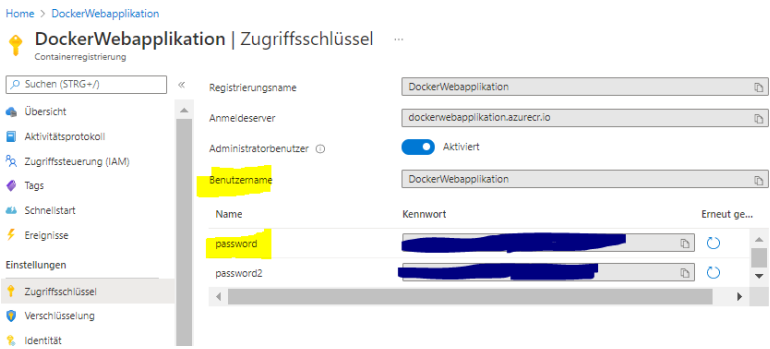
- *Kursiv teile in den Befehlen* = Stehen für individuelle Teile, dies hängt davon ab, wie man seine Objekte benannt an. z.B. mein Server heisst vielleicht anders als deiner.
- Einstellungen, die nicht im Bild zu sehen sind, wurden immer auf der Standardeinstellung gelassen
- Es wurde kein Portforwarding erstellt. D.H der korrekte Port muss angegeben werden beim Aufrufen über den Browser co.

Anleitung

ID	Beschreibung	Grafik
1	Azure → Icon Ressourcengruppen anwählen und dann auf Knop „Erstellen“ klicken	
2	Spezifikationen eingeben und erstellen	

3	<p>Containerregistrierungen -> erstellen -> Spezifikationen eingeben und erstellen klicken</p>	<p>Containerregistrierung erstellen ...</p> <p>Grundeinstellungen Netzwerk Verschlüsselung Tags Überprüfen + erstellen</p> <p>Azure Container Registry ermöglicht Ihnen das Erstellen, Speichern und Verwalten von Containerimages und Artefakten in einer privaten Registrierung für alle Arten von Containerbereitstellungen. Verwenden Sie Azure-Containerregistrierungen für Ihre vorhandenen Pipelines zur Containerentwicklung und -bereitstellung. Verwenden Sie Azure Container Registry-Aufgaben, um Containerimages in Azure bedarfsgesteuert zu erstellen, oder automatisieren Sie Builds, die durch Quellcodeaktualisierungen, Aktualisierungen eines Containerbasisimages oder durch Timer ausgelöst werden. Weitere Informationen</p> <p>Projektdetails</p> <p>Abonnement * <input type="text" value="Azure for Students"/></p> <p>Ressourcengruppe * <input type="text" value="VIPA"/> Neues Element erstellen</p> <p>Instanzendetails</p> <p>Registrierungsname * <input type="text" value="DockerWebapplikation"/> <input checked="" type="checkbox"/> <small>.azurecr.io</small></p> <p>Standort * <input type="text" value="Schweiz Norden"/></p> <p>Verfügbarkeitszonen <input type="checkbox"/> Aktiviert</p> <p><small>Verfügbarkeitszonen sind für Premium-Registrierungen und in Regionen aktiviert, die Verfügbarkeitszonen unterstützen. Weitere Informationen</small></p> <p>SKU * <input type="text" value="Standard"/></p>
4	<p>Ins Verzeichnis der Webapp navigieren → File Namens „dockerfile“ erstellen → Inhalt vom Bild einfügen.</p> <p>Achtung eventuell haben sie einen anderen Port definiert als ich. In diesem Fall ihren Port bei EXPOSE angeben.</p>	<pre> 1 FROM node:16 2 3 # Create app directory 4 WORKDIR /usr/src/app 5 6 # Install app dependencies 7 # A wildcard is used to ensure both package.json AND package-lock.json are copied 8 # where available (npm@5+) 9 COPY package*.json ./ 10 11 RUN npm install 12 # If you are building your code for production 13 # RUN npm ci --only=production 14 15 # Bundle app source 16 COPY . . 17 18 EXPOSE 8080 19 CMD ["node", "index.js"] </pre>

5	Im selben Verzeichnis File Namens „dockerignore“ erstellen → Inhalt vom Bild einfügen.	
6	Sicherstellen, dass im File Pacakge.json unter scripts "start": "node index.js" definiert ist.	
6	Im Terminal zum Ordner der Webapplikation navigieren	
7	Im Terminal den Befehl „docker build -t anmeldeserver/registrierungsname“ Die Daten dazu sind auf azure unter Einstellungen → Zugriffsschlüssel zu finden (beim Befehl alles Lowercase).	

8	Überprüfen, ob das Image erstellt wurde mit dem Befehl „docker image ls“	<pre>PS C:\Users\in19s\Desktop\basic_webapp> docker image ls</pre> <table><thead><tr><th>REPOSITORY</th><th>TAG</th><th>IMAGE ID</th><th>CREATED</th><th>SIZE</th></tr></thead><tbody><tr><td>dockerwebapplikation.azurecr.io/dockerwebapplikation</td><td>latest</td><td>4443dc254cb2</td><td>48 seconds ago</td><td>914MB</td></tr></tbody></table>	REPOSITORY	TAG	IMAGE ID	CREATED	SIZE	dockerwebapplikation.azurecr.io/dockerwebapplikation	latest	4443dc254cb2	48 seconds ago	914MB
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE								
dockerwebapplikation.azurecr.io/dockerwebapplikation	latest	4443dc254cb2	48 seconds ago	914MB								
9	Im Anmeldeserver einloggen.	<pre>PS C:\Users\in19s\Desktop\basic_webapp> docker login dockerwebapplikation.azurecr.io</pre>										
10	Nun muss im Terminal der Username und das Passwort eingegeben werden. Sie finden dies unter Zugriffsschlüssel nachdem sie Administratorbenutz er aktiviert haben.	 <pre>Username: DockerWebapplikation Password: Login Succeeded</pre>										
11	Das Docker Image mit dem Befehl „docker push repository“ hochladen	<pre>docker push dockerwebapplikation.azurecr.io/dockerwebapplikation</pre>										

12

Um zu kontrollieren ob das Pushen erfolgreich war, auf azure navigieren → Containerregistrierung → Dienste → Repositorys → auf Repository klicken.

The screenshot displays the Docker Web Application interface for managing container registries. The main heading is "DockerWebapplikation | Repositorys". Below this, there's a search bar with the placeholder "Suchen (STRG+/-)" and an "Aktualisieren" button. A left sidebar contains a navigation menu with options like "Übersicht", "Aktivitätsprotokoll", "Zugriffssteuerung (IAM)", "Tags", "Schnellstart", "Ereignisse", "Einstellungen" (with sub-items like "Zugriffsschlüssel", "Verschlüsselung", "Identität", "Netzwerk", "Sicherheit", "Sperrern"), and "Dienste" (with sub-items like "Repositorys" and "Webhooks"). The "Repositorys" section is active. The main content area shows a search bar "Suchen, um Repositorys zu filtern..." and a list of repositories. One repository, "dockerwebapplikation", is highlighted in yellow. Below this, there's a detailed view for the "dockerwebapplikation" repository, showing options to "Aktualisieren" or "Repository löschen", a "Zusammenfassung" section, and a table of tags. The table has columns for "Repository" (showing "dockerwebapplikation") and "Datum der letzten Aktual..." (with a redacted value). A search bar "Suchen, um Tags zu filtern..." is present, and the "latest" tag is listed.

13

Azure →
Containerinstanzen
→ Erstellen →
Spezifikationen
eingeben →
erstellen

Hier muss als
Imagequelle Azure
Container Registry
gewählt werden,
weil wir dort unser
Repository gepusht
haben.

Als Port den vorher
definierten Port
eingeben .

Containerinstanz erstellen ...

Basics Netzwerk Erweitert Tags Review + create

Mit Azure Container Instances (ACI) können Sie auf einfache und schnelle Weise Container in Azure ausführen, ohne Server verwalten oder sich mit neuen Tools vertraut machen zu müssen. ACIs ermöglichen eine sekundengenaue Abrechnung, um die Kosten für den Betrieb von Containern in der Cloud zu minimieren. [Weitere Informationen zu Azure Container Instances](#)

Project details

Wählen Sie das Abonnement aus, um bereitgestellte Ressourcen und Kosten zu verwalten. Verwenden Sie Ressourcengruppen wie z. B. Ordner zum Organisieren und Verwalten all Ihrer Ressourcen.

Subscription * ⓘ Azure for Students
Resource group * ⓘ VIPA
[Neues Element erstellen](#)

Details zum Container

Containername * ⓘ dockertestwebapp ✓
Region * ⓘ (Europe) Schweiz Norden
Imagequelle * ⓘ
☐ Schnellstartimages
☒ Azure Container Registry
☐ Docker Hub oder andere Registrierung
Registrierung * ⓘ DockerWebapplikation
Image * ⓘ dockerwebapplikation
Imagetag * ⓘ latest
Betriebssystemtyp Linux
Größe * ⓘ 1 vcpu, 1.5 GiB Arbeitsspeicher, 0 gpus
[Größe ändern](#)

Containerinstanz erstellen ...

Basics **Netzwerk** Erweitert Tags Review + create

Wählen Sie zwischen drei Netzwerkoptionen für Ihre Containerinstanz:

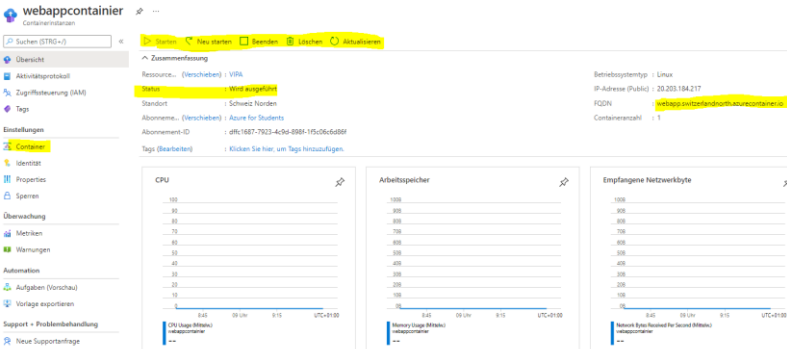
- Bei **Öffentlich** wird eine öffentliche IP-Adresse für Ihre Containerinstanz erstellt.
- Bei **Privat** können Sie ein neues oder vorhandenes virtuelles Netzwerk für Ihre Containerinstanz auswählen. Diese Option ist für Windows-Container noch nicht verfügbar.
- Bei **Kein** wird weder eine öffentliche IP-Adresse noch ein virtuelles Netzwerk erstellt. Sie können über die Befehlszeile weiterhin auf Ihre Containerprotokolle zugreifen.

Netzwerktyp ☒ Öffentlich ☐ Privat ☐ Keine

DNS-Namensbezeichnung ⓘ webapp
.switzerlandnorth.azurecontainer.io

Ports ⓘ

Ports	Portprotokoll
8080 ✓	TCP

<p>14</p>	<p>Azure → Ressourcegruppen → vorher erstellte Containerinstanz wählen (in meinem Fall: dockertestwebappc ontainer)</p> <p>Hier kann der Webapp Container verwaltet werden</p>	
<p>15</p>	<p>Nun kann die Webapp im Browser angerufen werden → FQDN kopieren und Port hinzufügen.</p>	